**2D array, pointers, strings**

1. Using 2D arrays, implement various matrix operations:
   1. Addition
   2. Subtraction
   3. Multiplication
   4. Transpose
   5. Row sum, given a row id
   6. Column sum, given a column id
   7. Diagonal sum (combining major and minor diagonals)
2. Given a square matrix, check if it is an **Identity matrix**. (Find out what an identity matrix is)
3. Implement functions for each of the operations mentioned in **Q.1 and Q2. Use appropriate parameters.**
4. Implement the functionality of Q.1 and Q.3 now with dynamic allocation. Assume the maximum number of rows is known beforehand, so you can use array of pointers (int\* matrix[MAXROW])
5. Now use 2D dynamic memory allocation (int \*\*matrix) for the above tasks.
6. Write code to take a list of names from console. Use dynamic memory allocation (char \*\*). After taking the input, can you sort the names?
7. Implement the following string related functions. Check out Appendix of your book to understand what these functions do. (These are already implemented in the string library of C in the string.h header). But you need to understand the workings and provide your own implementation:
   1. strlen
   2. strcat
   3. strcpy
   4. strrev
   5. strtok
   6. strstr
8. Given a string, check whether it is a palindrome or not.
9. Given 2 strings str1, str2, remove all occurrences of str2 from str1.
10. Now solve problems 7-9 using pointer arithmetic, avoiding array indexing.